REMARKS

I. General

Claims 1-54 are pending in the present application. The present Office Action (mailed June 13, 2006) raises the following issues:

- Claims 1, 9, 21, 41, 47 are rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter;
- Claims 2-8 are objected to because of informalities;
- Claims 1-54 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,781,897 to Chen et al. (hereinafter "*Chen*").

Applicant respectfully traverses the outstanding claim rejections raised in the current Office Action, and requests reconsideration and withdrawal thereof in light of the amendments and remarks presented herein.

II. Amendments

Claims 1-8, 21, 47, and 54 are amended herein. Claim 49 is canceled without prejudice, and new claim 55 is added herein. No new matter is added by these amendments and newly added claim.

More specifically, claim 1 is amended to recite that the data records are received into "a record processing module of a system" and that the record processing module performs the recited "generating" of a text-string. Claim 1 is further amended to recite that the generating is "for application in efficiently searching for desired ones of said data records". Thus, as discussed below, this expressly recites a practical application of the claim.

Dependent claims 2-8 are amended to change their preambles to correctly refer to the "text-generation method" that is introduced in claim 1.

Claim 21 is amended to recite that the computer program causes the processor to process the received data records to generate the text-string for each data record. Claim 21 is further amended to recite that such generation of a text-string is "for application in efficiently searching for desired ones of said data records". Thus, as discussed below, this expressly recites a practical application of the claim.

Claim 47 is amended herein to recite that the data structure is stored to a computer-readable medium and is usable by a processor for efficiently searching for desired data records of a database. Claim 47 is also amended to clarify that the recited text-string represents one or more of the recited data records.

Claim 54 is amended to correct a typographical error so that claim 54 correctly depends from the data structure of claim 47, rather than from claim 1.

III. Rejections Under 35 U.S.C. §101

Claims 1, 9, 21, 41, 47 are rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. Applicant respectfully traverses these rejections below.

35 U.S.C. §101 provides:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Thus, 35 U.S.C. §101 defines four categories of inventions that Congress deemed to be appropriate subject matter of a patent: processes, machines, manufactures, and compositions of matter. The latter three categories define "things" or "products" while the first category defines "actions". *See* 35 U.S.C. §100(b) ("The term 'process' means process, art, or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material.").

As the Supreme Court held, Congress chose the expansive language of 35 U.S.C. §101 so as to include "anything under the sun that is made by man." *Diamond v. Chakrabarty*, 447 U.S. 303, 308-09, 206 USPQ 193, 197 (1980). The Federal Circuit has embraced this perspective:

> The plain and unambiguous meaning of section 101 is that any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may be patented if it meets the requirements for patentability set for in Title 35, such as those found in sections 102, 103, and 112. The use of the expansive term "any" in section 101 represents Congress's intent not to place any restrictions on the subject matter for which a patent may be obtained beyond those specifically recited in section 101 and the other parts of Title 35 ... Thus, it is improper to read into section 101 limitations as to the subject matter that may be patented where the legislative history does not indicate that Congress clearly intended such limitations. *In re Alappat*, 33 F.3d 1526, 1542, 31 USPQ2d 1545, 1556 (Fed. Cir. 1994).

Accordingly, a complete definition of the scope of 35 U.S.C. §101, reflecting Congressional intent, is that any new and useful process, machine, manufacture, or composition of matter (or any new and useful improvement thereof) under the sun that is made by man is the proper subject matter of a patent.

The subject matter courts have found to be outside of, or exceptions to, the four statutory categories of invention is limited to abstract ideas, laws of nature, and natural phenomena. These three judicial exclusions recognize that subject matter that is not a practical application or use of an idea, a law of nature, or a natural phenomena is not patentable. *See, e.g., Mackay Radio & Telegraph Co. v. Radio Corp. of America*, 306 U.S. 86, 94, 40 USPQ 199, 202 (1939) ("While a scientific truth, or the mathematical expression of it, is not patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be."). The courts have held that a claim may not preempt ideas, laws of nature, or natural phenomena. Accordingly, one may not patent every "substantial practical application" of an idea, law of nature, or natural phenomena because such a patent "in practical effect be a patent on the [idea, law of nature or natural phenomena] itself." *Gottschalk v. Benson*, 409 U.S. 63, 71-72, 175 USPQ 673, 676 (1972).

Thus, the USPTO guidelines recommend that patent personnel follow the following procedure when evaluating a claim to determine whether it is directed to statutory subject matter (*see e.g.,* United States Patent and Trademark Office OG Notices: 22 November 2005, *Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility*):

A. Patent personnel should first determine whether the claim falls within at least one of the four enumerated categories.

B. If the claim falls within at least one of the four enumerated categories, the inquiry does not end, but rather patent personnel should determine whether the claim covers a judicial exception, as being directed to an abstract idea, law of nature, or natural phenomena.

C. If determined that the claim is directed to an abstract idea, law of nature, or natural phenomena, patent personnel should go on to determine whether the claim is directed to a practical application thereof. The claim is determined to be directed to a practical application if: a) the claimed invention "transforms" an article or physical object to a different state or thing, or b) the claimed invention otherwise produces a useful, concrete, and tangible result.

Claims directed to one of the four statutory categories which do not fall under a judicial exception (i.e., are not an abstract idea, law of nature, or natural phenomena), are proper under 35 U.S.C. §101. Also, claims directed to one of the four statutory categories which fall under a judicial exception and are directed to a practical application are also proper under 35 U.S.C. §101.

As discussed below, the claims of the present application are directed to one of the four statutory categories expressly recognized by section 101. Further, the Office Action fails to establish that the claims are directed to a judicial exception to the recognized categories. Additionally, even if one or more of the claims are directed to a judicial exception, the claims do not preempt such judicial exception but are instead directed to a practical application thereof.

Claims 1 and 9

Claims 1 and 9 are directed to "methods" (i.e., to a "text-generation method" and to a "search method," respectively), and thus fall within one of the four statutory categories of section 101. The Office Action fails to establish that the claims are directed to a section 101 judicial exception (i.e., an abstract idea, law of nature, or natural phenomena). The USPTO guidelines explain that if an Examiner determines that the claimed invention preempts a section 101 judicial exception, "the Examiner must identify the abstraction, law of nature, or natural phenomenon and explain why the claim covers every substantial practical application thereof." The Office Action fails to identify any such abstraction, law of nature, or natural phenomenon.

The Office Action appears to conclude that the claims are not limited to a practical application of a judicial exception without first identifying any judicial exception to which the claimed are directed. That is, the Office Action fails to identify any abstraction, law of nature, or natural phenomenon that the claims cover. Thus, Applicant respectfully submits that the Office Action fails to establish a prima facie case for rejecting the claims under 35 U.S.C. §101.

Further, as discussed below, the claims do not preempt an abstract idea, law of nature, or natural phenomena, but rather, to the extent that they are directed to any such judicial exception, they are directed to a practical application thereof. For instance, independent claim 1 recites, in part "generating, for application in efficiently searching for desired ones of said data records, a text-string for each data record" (emphasis added). Thus, claim 1, as amended herein, expressly recites a practical application of generating a text-string, i.e., for application in efficiently searching for desired ones of said data records. As such, even assuming that generating a text-string is an abstract idea, law of nature, or natural phenomena (again, the Office Action fails to identify any such judicial exception), claim 1 does not preempt all applications of such generating a text-string but instead is expressly directed to generating such a text-string "for application in efficiently searching for desired ones of said data records".

If the claim is directed to a practical application of the judicial exception (i.e., abstract idea, law of nature, or natural phenomena) producing a result that does not preempt the judicial

exception, then the claim meets the statutory requirement of 35 U.S.C. §101. As described above, claim 1 is directed to a practical application, and thus the rejection under 35 U.S.C. §101 should be withdrawn.

Independent claim 9 recites:

> A search method comprising:
> defining a first target value for each of one or more data fields within a database record structure of a database, wherein the database includes a plurality of data records;
> searching a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
> a value descriptor that defines the field value associated with the specific data field; and
> generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values.

The Office Action appears to conclude that claim 9 is not limited to a practical application of a judicial exception without first identifying any judicial exception to which the claim is directed. That is, the Office Action fails to identify any abstraction, law of nature, or natural phenomenon that claim 9 preempts. Thus, Applicant respectfully submits that the Office Action fails to establish a prima facie case for rejecting the claim 9 under 35 U.S.C. §101.

Further, claim 9 does not preempt an abstract idea, law of nature, or natural phenomena, but rather, to the extent that the claim is directed to any such judicial exception, it is directed to a practical application thereof. That is, claim 9 produces a concrete, tangible, and useful result. For instance, independent claim 9 recites, in part "generating a first result set". Thus, claim 9 expressly recites generating a first result set. Such a first result set provides a concrete, tangible, and useful result.

In addressing whether a claim is directed to a practical application M.P.E.P. §2106 provides the following example:

> For example, a computer process that simply calculates a mathematical algorithm that models noise is nonstatutory. However, a claimed process for digitally filtering noise employing the mathematical algorithm is statutory.

Analogously, if claim 9 were directed merely to mathematical algorithms used for comparing text strings (e.g., boolean operations, such as AND, OR, etc.) so as to preempt such mathematical algorithms it might be nonstatutory, but in the present case claim 9 is not directed to the underlying mathematical algorithms employed. Instead, claim 9 expressly recites "generating a first result set". Thus, while the first result set may be generated through the use of underlying mathematical algorithms employed by the computer, claim 9 does not preempt all applications of any given mathematical algorithm, but instead is directed to a practical application in that it recites "generating a first result set", thereby producing a tangible, concrete, and useful result.

In view of the above, the rejection of claims 1 and 9 under 35 U.S.C. §101 should be withdrawn.

Claim 21

The Office Action further rejects claim 21. Independent claim 21, as amended herein, recites:

> A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:
>
> receive data records, wherein each data record includes one or more data fields and a field value associated with each data field; and
>
> process the received data records to generate, for application in efficiently searching for desired ones of said data records, a text-string for each data record, wherein each text-string includes one or more text-based data descriptors, such that each data descriptor includes:
>
> a field descriptor that defines a specific data field within the data

record to which the text-string is related, and

a value descriptor that defines the field value associated with the specific data field.

M.P.E.P. §2106 explains that "When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized." This is clearly the case for claim 21 as the functional code for receiving data records and processing the received data records to generate a text-string are stored to a computer-readable medium. Thus, claim 21 is directed to proper statutory subject matter under 35 U.S.C. §101.

Further, the Office Action has failed to identify any abstract idea, law of nature, or natural phenomenon that claim 21 is believed to be directed. Further, claim 21 is directed to the practical application of generating a text-string for each data record in that it expressly recites "for application in efficiently searching for desired ones of said data records".

In view of the above, the rejection of claim 21 under 35 U.S.C. §101 should be withdrawn.

Claim 41

Claim 41 is directed to a "system" (i.e., a "searching system"), and thus falls within one of the four statutory categories of section 101 (e.g., "machine"). The Office Action fails to establish that the claim is directed to a section 101 judicial exception (i.e., an abstract idea, law of nature, or natural phenomena). The USPTO guidelines explain that if an Examiner determines that the claimed invention preempts a section 101 judicial exception, "the Examiner must identify the abstraction, law of nature, or natural phenomenon and explain why the claim covers every substantial practical application thereof." The Office Action fails to identify any such abstraction, law of nature, or natural phenomenon.

The Office Action appears to conclude that the claim is not limited to a practical application of a judicial exception without first identifying any judicial exception to which the claim is directed. That is, the Office Action fails to identify any abstraction, law of nature, or natural phenomenon that the claim covers. Thus, Applicant respectfully submits that the Office Action fails to establish a prima facie case for rejecting claim 41 under 35 U.S.C. §101.

Further, as discussed below, claim 41 is not directed to an abstract idea, law of nature, or natural phenomena. For instance, independent claim 41 recites:

> A searching system comprising:
> a server system including a computer processor and associated memory, the server system having a database that includes a plurality of data records;
> wherein the server system is configured to:
> define a first target value for each of one or more data fields within a database record structure of the database;
> search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
> a value descriptor that defines the field value associated with the specific data field; and
> generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values.

Thus, claim 41 is directed to a system that comprises the various recited elements, such as a server system that includes a computer processor and associated memory. It is unclear what abstraction, law of nature, or natural phenomenon the Examiner believes is encompassed by such system (again, the Office Action fails to identify any such abstraction, law of nature, or natural phenomenon).

In view of the above, the rejection of claim 41 under 35 U.S.C. §101 should be withdrawn.

Claim 47

The Office Action further rejects claim 47. Independent claim 47, as amended herein, recites:

> A data structure stored to a computer-readable medium and usable by a processor for efficiently searching for desired data records of a database, said data structure comprising:
> a database including a plurality of data records, wherein each data record includes one or more data fields, and a field value is associated with each data field;
> a text-string representing one or more of said data records, wherein each text-string includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
> a value descriptor that defines the field value associated with the specific data field.

M.P.E.P. §2106(IV)(B)(1)(a) explains that "a claimed computer-readable medium encoded with a data structure defines structure and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory." *See In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory), as cited in M.P.E.P. §2106(IV)(B)(1). This is clearly the case for claim 47 as the recited data structure is recited as being "stored to a computer-readable medium and usable by a processor for efficiently searching for desired data records of a database". Thus, claim 41 is directed to proper statutory subject matter under 35 U.S.C. §101.

Further, the Office Action has failed to identify any abstract idea, law of nature, or natural phenomenon that claim 47 is believed to be directed. Further, claim 47 is directed to the practical application of efficiently searching for desired data records of a database.

In view of the above, the rejection of claim 47 under 35 U.S.C. §101 should be withdrawn.

IV. Claim Objections

Claims 2-8 are objected to because of informalities. In view of the amendments discussed above, the informalities have been resolved, and thus the objections should be withdrawn.

V. Rejections Under 35 U.S.C. §102(b) over *Chen*

Claims 1-54 are rejected under 35 U.S.C. §102(b) as being anticipated by *Chen*. To anticipate a claim under 35 U.S.C. §102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. As discussed below, Applicant respectfully traverses this rejection because *Chen* fails to teach all elements of the claims.

Independent Claim 1

Claim 1 recites:

> A text-generation method comprising:
> receiving, into a record processing module of a system, data records, wherein each data record includes one or more data fields and a field value associated with each data field; and
> said record processing module generating, for application in efficiently searching for desired ones of said data records, a text-string for each data record, wherein each text-string includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related; and
> a value descriptor that defines the field value associated with the specific data field. (Emphasis added).

Chen fails to teach all elements of claim 1. For instance, as discussed below, *Chen* does not teach "receiving, into a record processing module of a system, data records" and "said record processing module generating ... a text-string for each data record", as recited by claim 1.

In general, *Chen* is directed to "a method and system for executing searches within a computer system." Col. 1, lines 9-11 of *Chen*. *Chen* recognizes that inefficiency in searching for a given term in records of a database has arisen in traditional computer systems because the searching technique has required first copying each record to be searched from secondary storage into primary storage for processing by the processor. For instance, *Chen* explains at col. 1, lines 33-67 the following:

> Computer storage media form a storage hierarchy that includes two main categories--primary storage and secondary storage. Primary storage includes storage media that can be operated on directly by a central processing unit (CPU), such as a system main memory and/or a smaller but faster cache memory. Primary storage usually provides faster access to data but is of limited storage capacity. Secondary storage includes magnetic disks, tapes, and drums that are usually of larger capacity, but provide slower access to data than the primary storage.
>
> In performing database transactions, one of the most common operations is to search for a string of characters through one or several database tables within a database. For example, one could make a query for all records in a database table where a selected field includes a primary key of "Snoopy." When searching for the matching records, each and every record in the database table must be read and compared to the character string "Snoopy." If there is a match, the record will be selected. On the other hand, the record will be discarded if there is no match. Typically, data in secondary storage cannot be processed directly by the CPU; thus each record must be copied into primary storage in the system in which the comparison is performed. However, by reading a record from a disk into system main memory through several layers of I/O bus and system bus before the comparison of character string(s) can be performed, precious system resources (i.e., I/O bus, system bus, memory, and CPU time) will be wasted if the record being processed is to be discarded eventually. Unfortunately, the number of records that are discarded is often greater than the number of records that are selected in this kind of database transaction. Consequently, it would be desirable to provide a method to perform record searches more efficiently in a database within a computer system.

Thus, *Chen* proposes a system that "includes a main processor, a main memory, and a peripheral storage device having a secondary processor." Col. 2, lines 13-15 of *Chen*. Thus, "a command block specifying a search string for record searching in at least one database table of the database is prepared", and such "command block is issued from the main processor to the secondary processor within the peripheral storage device of the computer system". Col. 2, lines

15-21 of *Chen*. The peripheral storage device's secondary processor "is then utilized to read the database table(s) into a memory within the peripheral storage device, in response to receipt of the command block", and "the search string in the command block is compared to each record of the database table(s) within the memory of the peripheral storage device to identify all the records therein which contain the search string." Col. 2, lines 21-28 of *Chen*. Then, the matching records that are determined by the peripheral storage device as containing the search string are moved from the memory of the peripheral storage device to the main memory within the computer system. *See* col. 2, lines 28-33 of *Chen*. In this manner, the copying of each record to be searched into the main memory of the computer system is avoided by instead copying the records into memory of the peripheral storage device and using the peripheral storage device's secondary processor to search the database records for a search string.

Chen does not teach "receiving, into a record processing module of a system, data records" and "said record processing module generating ... a text-string for each data record", as recited by claim 1. *Chen* does not process the database records to generate a text-string for each record, but instead may receive into a peripheral storage device a search string (that is contained in a command block) and then search database records for such search string. No module in *Chen* generates a text-string for each database record. For instance, *Chen* does not teach that the search string contained in the command block is generated from processing of the database records, but instead it appears that the search string may be a user input search string, such as the word "Snoopy", which is then communicated to a peripheral storage device in a command block, wherein the peripheral storage device can then search database records for the search string. Again, in no case does *Chen* provide any teaching of a record processing module generating a text-string for each data record, but instead merely describes searching database records for a text-string that is included in a command block received by a peripheral storage device.

Thus, *Chen* does not anticipate claim 1 because it fails to teach at least the above-identified element of claim 1.

Independent Claim 9

Claim 9 recites:

> A search method comprising:
> defining a first target value for each of one or more data fields within a database record structure of a database, wherein the database includes a plurality of data records;
> searching a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
> a value descriptor that defines the field value associated with the specific data field; and
> generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values.

Chen fails to teach all elements of claim 9. For instance, as discussed further below, *Chen* fails to teach "searching a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors" and "generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values", as recited by claim 9.

As discussed above with claim 1, *Chen* proposes a system in that includes a main processor, a main memory, and a peripheral storage device having a secondary processor. Database records are copied into the peripheral storage device, and a command block specifying a search string for record searching may be issued from the main processor to the peripheral storage device's secondary processor. The secondary processor is then utilized to compare the search string to each record of the database table(s) that are copied into the memory of the peripheral storage device to identify all the records therein which contain the search string.

Chen does not teach "searching a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors" and "generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values", as recited by claim 9. Rather than searching a plurality of text-strings that is each associated with one of the data records, *Chen* expressly teaches copying data records into memory of a peripheral data storage device and searching the data records themselves for a search string received in a command block (e.g., for the search string "Snoopy"). That is, *Chen* does not teach searching a plurality of text-strings that are associated with data records, but instead teaches searching the data records themselves.

Further, because *Chen* does not teach searching text-strings that are associated with data records, *Chen* does not generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one target value. Instead, *Chen* directly searches the database records for a received search string (target value), such "Snoopy", to determine those records that match the search string.

Thus, *Chen* does not anticipate claim 9 because it fails to teach at least the above-identified elements of claim 9.

Independent Claim 21

Claim 21 recites:

> A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:
> receive data records, wherein each data record includes one or more data fields and a field value associated with each data field; and
> process the received data records to generate, for application in efficiently searching for desired ones of said data records, a text-string for each data record, wherein each text-string includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data

record to which the text-string is related, and

a value descriptor that defines the field value associated with the specific data field. (Emphasis added).

Chen fails to teach all elements of claim 21. For instance, as discussed below, *Chen* does not teach processing received data records to generate a text-string for each data record. That is, as discussed above with claim 1, *Chen* does not process the database records to generate a text-string for each record, but instead may receive into a peripheral storage device a search string (that is contained in a command block) and then search database records for such search string. *Chen* does not teach processing received database records to generates a text-string for each database record. For instance, *Chen* does not teach that the search string contained in the command block is generated from processing of the database records, but instead it appears that the search string may be a user input search string, such as the word "Snoopy", which is then communicated to a peripheral storage device in a command block, wherein the peripheral storage device can then search database records for the search string. Again, in no case does *Chen* provide any teaching of processing data records to generate a text-string for each data record, but instead merely describes searching database records for a text-string that is included in a command block received by a peripheral storage device.

Thus, *Chen* does not anticipate claim 21 because it fails to teach at least the above-identified element of claim 21.

Independent Claim 29

Claim 29 recites:

A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:

define a first target value for each of one or more data fields within a database record structure of a database, wherein the database includes a plurality of data records;

search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors,

such that each data descriptor includes:

a field descriptor that defines a specific data field within the data record to which the text-string is related, and

a value descriptor that defines the field value associated with the specific data field; and

generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values.

Chen fails to teach all elements of claim 29. For instance, as discussed further below, *Chen* fails to teach a computer program to cause a processor to "search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors" and "generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values", as recited by claim 29.

As discussed above with claim 1, *Chen* proposes a system in that includes a main processor, a main memory, and a peripheral storage device having a secondary processor. Database records are copied into the peripheral storage device, and a command block specifying a search string for record searching may be issued from the main processor to the peripheral storage device's secondary processor. The secondary processor is then utilized to compare the search string to each record of the database table(s) that are copied into the memory of the peripheral storage device to identify all the records therein which contain the search string.

Chen does not teach a computer program to cause a processor to "search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors" and "generating a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values", as recited by claim 29. Rather than searching a plurality of text-strings that is each associated with one of the data records, *Chen* expressly teaches copying data records into memory of a peripheral data storage device and searching the data records themselves for a search string received in a command block (e.g., for the search string "Snoopy"). That is, *Chen*

does not teach searching a plurality of text-strings that are associated with data records, but instead teaches searching the data records themselves.

Further, because *Chen* does not teach searching text-strings that are associated with data records, *Chen* does not generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one target value. Instead, *Chen* directly searches the database records for a received search string (target value), such "Snoopy", to determine those records that match the search string.

Thus, *Chen* does not anticipate claim 29 because it fails to teach at least the above-identified elements of claim 29.

Independent Claim 41

Claim 41 recites:

> A searching system comprising:
> a server system including a computer processor and associated memory, the server system having a database that includes a plurality of data records;
> wherein the server system is configured to:
> define a first target value for each of one or more data fields within a database record structure of the database;
> search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-based data descriptors, such that each data descriptor includes:
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
> a value descriptor that defines the field value associated with the specific data field; and
> generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values.

Chen fails to teach all elements of claim 41. For instance, as discussed further below, *Chen* fails to teach a server system that is configured to "search a plurality of text-strings, wherein each text string is associated with one of the data records and includes one or more text-

based data descriptors" and "generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one of the first target values", as recited by claim 41.

As discussed above with claim 1, *Chen* proposes a system in that includes a main processor, a main memory, and a peripheral storage device having a secondary processor. Database records are copied into the peripheral storage device, and a command block specifying a search string for record searching may be issued from the main processor to the peripheral storage device's secondary processor. The secondary processor is then utilized to compare the search string to each record of the database table(s) that are copied into the memory of the peripheral storage device to identify all the records therein which contain the search string.

Rather than searching a plurality of text-strings that is each associated with one of the data records, *Chen* expressly teaches copying data records into memory of a peripheral data storage device and searching the data records themselves for a search string received in a command block (e.g., for the search string "Snoopy"). That is, *Chen* does not teach searching a plurality of text-strings that are associated with data records, but instead teaches searching the data records themselves.

Further, because *Chen* does not teach searching text-strings that are associated with data records, *Chen* does not generate a first result set by identifying one or more text-strings that include a value descriptor that is essentially equivalent to at least one target value. Instead, *Chen* directly searches the database records for a received search string (target value), such "Snoopy", to determine those records that match the search string.

Thus, *Chen* does not anticipate claim 41 because it fails to teach at least the above-identified elements of claim 41.

Independent Claim 47

Claim 47 recites:

> A data structure stored to a computer-readable medium and usable by a processor for efficiently searching for desired data records of a database, said data structure comprising:
>
> a database including a plurality of data records, wherein each data record includes one or more data fields, and a field value is associated with each data field;
>
> a text-string representing one or more of said data records, wherein each text-string includes one or more text-based data descriptors, such that each data descriptor includes:
>
> a field descriptor that defines a specific data field within the data record to which the text-string is related, and
>
> a value descriptor that defines the field value associated with the specific data field. (Emphasis added).

Chen fails to teach all elements of claim 47. For instance, as discussed below, *Chen* does not teach a text-string representing one or more data records. That is, as discussed above with claim 1, *Chen* does not teach a text-string representing database records, but instead teaches that the database records themselves are searched. *Chen* teaches that a peripheral storage device may receive a search string, such as "Snoopy" (that is contained in a command block), and then the peripheral storage device searches database records for such search string. *Chen* does not a data structure comprising a text -string representing one or more of the database records. Rather, *Chen* merely compares a received search string (e.g., "Snoopy") against the database records themselves. The search string that is contained in a command block in *Chen* does not represent one or more data records, but is instead used for comparison against database records in order to identify matching records. In no case does *Chen* teach a text-string that represents one or more data records.

Thus, *Chen* does not anticipate claim 47 because it fails to teach at least the above-identified element of claim 47.

Dependent Claims

Claims 2-8, 10-20, 22-28, 30-40, 42-46, 48, and 50-55 each depend from one of independent claims 1, 9, 21, 29, 41, and 47, and are thus likewise believed to be allowable at least based on their dependency from their respective independent claim for the reasons discussed above. Accordingly, Applicant respectfully requests that the rejection of claims 2-8, 10-20, 22-28, 30-40, 42-46, 48, and 50-55 also be withdrawn.

Conclusion

In view of the above, Applicant believes the pending application is in condition for allowance.

The required fee for this response is enclosed. If any additional fee is due, please charge Deposit Account No. 50-3948, under Order No. 66729/P029US from which the undersigned is authorized to draw.

Dated: October 13, 2006

Respectfully submitted,

By _____
Jody C. Bishop
Registration No.: 44,034
FULBRIGHT & JAWORSKI L.L.P.
2200 Ross Avenue, Suite 2800
Dallas, Texas 75201-2784
(214) 855-8007
(214) 855-8200 (Fax)
Attorney for Applicant